Reasoning About Strategies[§]

Fabio Mogavero[*†] [1], **Aniello Murano**[*] [1], **and Moshe Y. Vardi**[‡] [2]

1 **Universitá degli Studi di Napoli "Federico II", I-80126 Napoli, Italy.**
`{mogavero, murano}@na.infn.it`
2 **Rice University, Department of Computer Science, Houston, TX 77251-1892, U.S.A.**
`vardi@cs.rice.edu`

Abstract

In open systems verification, to formally check for reliability, one needs an appropriate formalism to model the interaction between open entities and express that the system is correct no matter how the environment behaves. An important contribution in this context is given by *modal logics for strategic ability*, in the setting of *multi-agent games*, such as ATL, ATL*, and the like. Recently, Chatterjee, Henzinger, and Piterman introduced *Strategy Logic*, which we denote here by $\mathrm{SL}_{\mathrm{CHP}}$, with the aim of getting a powerful framework for reasoning explicitly about strategies. $\mathrm{SL}_{\mathrm{CHP}}$ is obtained by using first-order quantifications over strategies and it has been investigated in the specific setting of two-agents turned-based game structures where a non-elementary model-checking algorithm has been provided. While $\mathrm{SL}_{\mathrm{CHP}}$ is a very expressive logic, we claim that it does not fully capture the strategic aspects of multi-agent systems.

In this paper, we introduce and study a more general strategy logic, denoted SL, for reasoning about strategies in multi-agent concurrent systems. We prove that SL strictly includes $\mathrm{SL}_{\mathrm{CHP}}$, while maintaining a decidable model-checking problem. Indeed, we show that it is 2EXPTIME-COMPLETE, thus not harder than that for ATL* and a remarkable improvement of the same problem for $\mathrm{SL}_{\mathrm{CHP}}$. We also consider the satisfiability problem and show that it is undecidable already for the sub-logic $\mathrm{SL}_{\mathrm{CHP}}$ under the concurrent game semantics.

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2010.133

1 Introduction

In system design, *model checking* is a well-established formal method that allows to automatically check for global system correctness [4, 19, 5]. In such a framework, in order to check whether a system satisfies a required property, we express the system in a formal model (such as a *Kripke* structure), specify the property with a temporal-logic formula (such as LTL [18], CTL [4], or CTL* [7]), and check formally that the model satisfies the formula. In the last decade, interest has arisen in analyzing the behavior of individual components and sets of components in systems with several components. This interest has started in reactive systems, which are systems that interact continually with their environments. In *module checking* [14] the system is modeled as a module that interacts with its environment and correctness means that a desired property holds with respect to all such interactions.

Starting from the study of module checking, researchers have looked for logics focusing on strategic behavior of agents in multi-agent systems [1, 16, 10]. One of the most important development in this field is *Alternating-Time Temporal Logic* (ATL*, for short), introduced by Alur, Henzinger,

[§] Part of this research was done while the authors were visiting the Hebrew University.
[*] Work partially supported by MIUR PRIN Project n.2007-9E5KM8, ESF GAMES Project, and Vigevani Research Project Prize 2010.
[†] Part of this research was done while visiting the Rice University.
[‡] Work supported in part by NSF grants CCF-0613889, CCF-0728882, and CNS 1049862, by BSF grant 9800096, and by gift from Intel.

IARCS Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010).
Editors: Kamal Lodaya, Meena Mahajan; pp. 133–144

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and Kupferman [1]. ATL* allows reasoning about strategies for agents with temporal goals. Formally, it is obtained as a generalization of CTL* in which the path quantifiers, "E" (*there exists*) and "A" (*for all*) are replaced with "*strategic modalities*" of the form $\langle\langle A \rangle\rangle$ and $[\![A]\!]$, where A is a set of *agents* (a.k.a. *players*). Strategic modalities over agent sets are used to express cooperation and competition among agents in order to achieve certain goals. In particular, these modalities express selective quantifications over those paths that are the results of infinite games between the coalition and its complement. ATL* formulas are interpreted over *game structures*, which model interacting processes. Given a game structure $\mathcal{S}$ and a set A of agents, the ATL* formula $\langle\langle A \rangle\rangle \psi$ is satisfied at a state s of $\mathcal{S}$ if there is a *strategy* for the agents in A such that, no matter the strategy that is executed by agents not in A, the resulting outcome of the interaction satisfies ψ at s. Thus, ATL* can express properties related to the interaction among agents, while CTL* can only express property of the global system. As an example, consider the property "processes α and β cooperate to ensure that a system (having more than two processes) never enters a fail state". This property can be be expressed by the ATL* formula $\langle\langle \{\alpha, \beta\} \rangle\rangle \mathsf{G} \neg \mathit{fail}$, where G is the classical temporal modality "*globally*". CTL*, in contrast, cannot express this property [1]. Indeed, CTL* can only say whether the set of all agents can or cannot prevent the system from entering a fail state. The price that one pays for the expressiveness of ATL* is increased complexity; both model checking and satisfiability checking are 2EXPTIME-COMPLETE [1, 20].

Despite its powerful expressiveness, ATL* suffers of the strong limitation that strategies are treated only implicitly, through modalities that refer to games between competing coalitions. To overcome this problem, Chatterjee, Henzinger, and Piterman introduced Strategy Logic ($\mathrm{SL}_{\mathrm{CHP}}$, for short) [3], a logic that treats strategies in two-player games as explicit first-order objects. In $\mathrm{SL}_{\mathrm{CHP}}$, the ATL* formula $\langle\langle \alpha \rangle\rangle \psi$ becomes $\exists x. \forall y. \psi(x, y)$, i.e., "there exists a player-α strategy x such that for all player-β strategies y, the unique infinite path resulting from the two players following the strategies x and y satisfies the property ψ". The explicit treatment of strategies in $\mathrm{SL}_{\mathrm{CHP}}$ allows to state many properties not expressible in ATL*. In particular, it is shown in [3] that ATL* corresponds to the proper one-alternation fragment of $\mathrm{SL}_{\mathrm{CHP}}$. Chatterjee et al. have shown that the model-checking problem for $\mathrm{SL}_{\mathrm{CHP}}$ is decidable, although only a non-elementary algorithm for it, both in the size of the system and the size formula, has been provided, leaving as open the question whether an algorithm with a better complexity exists or not. The question about the decidability of satisfiability checking for $\mathrm{SL}_{\mathrm{CHP}}$ was also left open in [3].

While the basic idea exploited in [3] to quantify over strategies, and thus to commit agent explicitly to certain strategies, turns out to be very powerful, as discussed above, the logic $\mathrm{SL}_{\mathrm{CHP}}$ introduced there has been defined and investigated only under the weak framework of two-players and turn-based games. Also, the specific syntax considered for $\mathrm{SL}_{\mathrm{CHP}}$ allows only a weak kind of strategy commitment. For example, $\mathrm{SL}_{\mathrm{CHP}}$ does not allow different players to share, in different contexts, the same strategy. These considerations, as well as all questions left open about $\mathrm{SL}_{\mathrm{CHP}}$, have led us to introduce and investigate a new *Strategy Logic*, denoted SL, as a more general framework than $\mathrm{SL}_{\mathrm{CHP}}$, for explicit reasoning about strategies in multi-player concurrent game structures. Syntactically, SL extends LTL by means of two *strategy quantifiers*, the existential $\langle\langle x \rangle\rangle$ and the universal $[\![x]\!]$, and an *agent binding* (α, x), where α is an agent and x is variable. Intuitively, these elements can be respectively read as *"there exists a strategy x"*, *"for all strategies x"*, and *"bind agent* α *to the strategy associated with x"*. For example, in a system with three agents α, β, γ, the previous ATL* formula $\langle\langle \{\alpha, \beta\} \rangle\rangle \mathsf{G} \neg \mathit{fail}$ can be expressed by the SL formula $\langle\langle x \rangle\rangle \langle\langle y \rangle\rangle [\![z]\!](\alpha, x)(\beta, y)(\gamma, z)(\mathsf{G} \neg \mathit{fail})$. The variables x and y are used to select two strategies for the agents α and β, respectively, and z is used to select all strategies for agent γ such that the composition of all these strategies results in a play where *fail* is never meet. Note that we can also require (by means of agent binding) that agents α and β share the same strategy, using the formula $\langle\langle x \rangle\rangle [\![z]\!](\alpha, x)(\beta, x)(\gamma, z)(\mathsf{G} \neg \mathit{fail})$. We can also

vary the structure of the game by changing the way the quantifiers alternate, for example, in the formula $\langle\langle x\rangle\rangle[[z]]\langle\langle y\rangle\rangle(\alpha,x)(\beta,y)(\gamma,z)(\mathsf{G}\,\neg \mathit{fail})$. In this case, x remains uniform w.r.t. z, but y becomes dependent on z. The last two examples show that SL is a proper extension of both ATL* and $\mathrm{SL}_{\mathrm{CHP}}$. It is worth noting that the pattern of modal quantifications over strategies and binding to agents can be extended to other logics than LTL, such as the linear μ-CALCULUS. In fact, the use of LTL here is only a matter of simplicity in presenting our framework, and changing the embedded temporal logic involves only few side-changes in the decision procedures.

As a main result in this paper, we show that the model-checking problem for SL is decidable and precisely PTIME in the size of the model and 2EXPTIME-COMPLETE in the size of the specification, thus not harder than that for ATL*. Remarkably, this result improves significantly the complexity of the model-checking problem for $\mathrm{SL}_{\mathrm{CHP}}$, for which only a non-elementary upper-bound was known [3]. The lower bound for the addressed problem immediately follows from ATL*, which SL includes. For the upper bound, we follow an *automata-theoretic approach* [13], by reducing the decision problem for the logic to the emptiness problem of automata. To this aim, we use *alternating Parity tree automata*, which are *alternating tree automata* (see [8], for a survey) along with a Parity acceptance condition [15]. Due to the exponential size of the required automaton and the EXPTIME complexity required for checking its emptiness, we get the desired 2EXPTIME upper bound.

As another important issue in this paper, we address the satisfiability problem for SL. By using a reduction from the *recurrent domino problem*, we show that this problem is highly undecidable, and in fact Σ^1_1-HARD, (i.e., it is not computably enumerable). Interestingly, the reduction we propose also holds for $\mathrm{SL}_{\mathrm{CHP}}$, under the concurrent game semantics. Thus, we show that in this setting also $\mathrm{SL}_{\mathrm{CHP}}$ is highly undecidable, while it remains an open question whether it is decidable or not in the turned-based framework. A key point to prove the undecidability of SL has been to show that this logic lacks of the bounded-tree model property, which does hold for ATL* [20].

Since the rise of temporal and modal program logics in the mid-to-late 1970s, we have learned to expect such logics to have a decidable satisfiability problem. In the context of temporal logic, decidability results were extended from LTL, to CTL* and to ATL*. SL deviates from this pattern. It has a decidable model-checking problem, but an undecidable satisfiability problem. In this, it is similar to first-order logic. The decidability of model checking for first-order logic is the foundation for query evaluation in relational databases, and undecidability of satisfiability is a challenge we need to contend with. At the same time, it is clear that SL has nontrivial fragments, for example, ATL*, which do have a decidable satisfiability problem. Identifying larger fragments of SL with a decidable satisfiability problem is an important research problem.

Related Work Several works have focused on extensions of ATL* to incorporate more powerful strategic constructs. Among them, we recall the logics *Alternating-Time μ-calculus* [1], *Game Logic* [1], $\mathrm{QD}\mu$ [17], and some extensions of ATL* considered in [2]. AMC and $\mathrm{QD}\mu$ are intrinsically different from SL (as well as $\mathrm{SL}_{\mathrm{CHP}}$ and ATL*) as they are obtained by extending the propositional μ-calculus [11] with strategic modalities. GL is strictly included in $\mathrm{SL}_{\mathrm{CHP}}$, but does not use any explicit treatment of strategies. Also the extensions of ATL* considered in [2] do not use any explicit treatment of strategies. Rather, they consider restrictions on the memory for strategy quantifiers. Thus, all the above logics are different from SL, which aims it at being a minimal but powerful logic to reason about strategic behavior in multi-agent systems.

Due to the lack of space, all proofs are omitted and reported in the full version.

2 Preliminaries

A *concurrent game structure* (CGS, for short) is a tuple $\mathcal{G} \triangleq \langle \mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \lambda, \tau, s_0\rangle$, where AP and Ag are finite non-empty sets of *atomic propositions* and *agents*, Ac and St are enumerable non-empty

sets of *actions* and *states*, $s_0 \in \mathrm{St}$ is a designated *initial state*, and $\lambda : \mathrm{St} \to 2^{\mathrm{AP}}$ is a *labeling* function that maps each state to the set of atomic propositions true in that state. Let $\mathrm{Dc} \triangleq \mathrm{Ac}^{\mathrm{Ag}}$ be the set of *decisions*, commonly known as *action profiles*, that are functions from Ag to Ac representing the choices of an action for each agent. Then, $\tau : \mathrm{St} \times \mathrm{Dc} \to \mathrm{St}$ is a *transition* function mapping a state and a decision to a state. Intuitively, CGSs provide a generalization of *labeled transition systems*, modeling multi-agent systems, viewed as *multi-player games* in which players perform *concurrent actions*, chosen strategically as a function of the history of the game. Note that elements in St are not global states of the system, but states of the environment in which the agents operate. Thus, they can be viewed as states of the game, which do not include the local states of the agents. By $|\mathcal{G}| \triangleq |\mathrm{St}| \cdot |\mathrm{Dc}|$ we denote the *size* of $\mathcal{G}$, which also corresponds to the size $|\mathsf{dom}(\tau)|$ of the transition function τ. If the set of actions is finite, i.e., $b = |\mathrm{Ac}| < \infty$, we say that $\mathcal{G}$ is *b-bounded*, or simply *bounded*. If both the sets of actions and states are finite, we say that $\mathcal{G}$ is *finite*. It is immediate to note that $\mathcal{G}$ is finite iff it has a finite size.

A *track* (resp., *path*) is a finite (resp., an infinite) sequence of states $\rho \in \mathrm{St}^*$ (resp., $\pi \in \mathrm{St}^\omega$) such that, for all $0 \leq i < |\rho| - 1$ (resp., $i \in \mathbb{N}$), there exists $\mathsf{d} \in \mathrm{Dc}$ such that $\rho_{i+1} = \tau(\rho_i, \mathsf{d})$ (resp., $\pi_{i+1} = \tau(\pi_i, \mathsf{d})$). Intuitively, tracks and paths of a CGS $\mathcal{G}$ are legal sequences of reachable states in $\mathcal{G}$ that can be seen as a description of the possible *outcomes* of the game modeled by $\mathcal{G}$. A track ρ is said *non-trivial* iff $|\rho| > 0$. We use $\mathrm{Trk} \subseteq \mathrm{St}^+$ (resp., $\mathrm{Pth} \subseteq \mathrm{St}^\omega$) to indicate the sets of all non-trivial tracks (resp., paths). By $\mathsf{fst}(\rho) \triangleq \rho_0$ (resp., $\mathsf{lst}(\rho) \triangleq \rho_{|\rho|-1}$), we denote the *first* (resp., *last*) state of the track ρ and, by $\rho_{\leq i}$, we denote the *prefix* up to the state of index $i < |\rho|$ of the track ρ, i.e., the track built by the first $i+1$ states $\rho_0, \ldots, \rho_i$ of ρ. The notations of first and prefix apply also to paths.

A *strategy* is a partial function $\mathsf{f} : \mathrm{Trk} \rightharpoonup \mathrm{Ac}$, non associated to any particular agent, mapping each non-trivial track in its domain to an action. Intuitively, a strategy is a *plan* for an agent that contains all choices of moves as a function of the history of the current outcome. We use Str to indicate the sets of all strategies. For a state s, we say that f is *s-total* iff it is defined on all non-trivial tracks starting in s, i.e., $\mathsf{dom}(\mathsf{f}) = \{\rho \in \mathrm{Trk} \mid \mathsf{fst}(\rho) = s\}$. For a track $\rho \in \mathsf{dom}(\mathsf{f})$, by f_ρ we denote the *translation* of f along ρ, i.e., the $\mathsf{lst}(\rho)$-total strategy such that $\mathsf{f}_\rho(\mathsf{lst}(\rho) \cdot \rho') = \mathsf{f}(\rho \cdot \rho')$, for all $\mathsf{lst}(\rho) \cdot \rho' \in \mathsf{dom}(\mathsf{f}_\rho)$.

Let Var be a fixed set of *variables*. An *assignment* is a partial function $\chi : \mathrm{Ag} \cup \mathrm{Var} \rightharpoonup \mathrm{Str}$ mapping every agent and variable to a strategy. An assignment χ is *complete* iff $\mathrm{Ag} \subseteq \mathsf{dom}(\chi)$. We use Asg to indicate the sets of all assignments. For a state s, we say that χ is *s-total* iff all strategies $\chi(l)$ are s-total too, for $l \in \mathsf{dom}(\chi)$. Let ρ be a track and χ be an $\mathsf{fst}(\rho)$-total assignment. By χ_ρ we denote the *translation* of χ along ρ, i.e., the $\mathsf{lst}(\rho)$-total assignment with $\mathsf{dom}(\chi_\rho) = \mathsf{dom}(\chi)$, such that $\chi_\rho(l) = \chi(l)_\rho$, for all $l \in \mathsf{dom}(\chi)$. Intuitively, the translation χ_ρ is the update of all strategies contained into the assignment χ, after the history of the game becomes ρ. Let χ be an assignment, a be an agent, x be a variable, and f be a strategy. Then, by $\chi[a \mapsto \mathsf{f}]$ and $\chi[x \mapsto \mathsf{f}]$ we denote, respectively, the new assignments defined on $\mathsf{dom}(\chi) \cup \{a\}$ and $\mathsf{dom}(\chi) \cup \{x\}$ that return f on a and x and are equal to χ on the remaining part of its domain. Note that, if χ and f are s-total, $\chi[a \mapsto \mathsf{f}]$ and $\chi[a \mapsto \mathsf{f}]$ are s-total, too.

Finally, a path π starting in a state s is a *play* w.r.t. a complete s-total assignment χ ((χ, s)-*play*, for short) iff, for all $i \in \mathbb{N}$, it holds that $\pi_{i+1} = \tau(\pi_i, \mathsf{d})$, where $\mathsf{d}(a) = \chi(a)(\pi_{\leq i})$, for all $a \in \mathrm{Ag}$. Note that there is a unique (χ, s)-play. Intuitively, a play is the outcome of the game determined by all the agent strategies participating to the game.

In the sequel of the paper, we use the Greek letters "α, β, γ" with indexes to indicate specific agents of a CGS, while we use the Latin letter "a" as a meta-variable on the agents themselves.

3 Strategy Logic

In this section, we formally introduce SL and discuss its main properties. In particular, we show that it does not have the bounded-tree model property.

Syntax. SL syntactically extends LTL by means of two *strategy quantifiers*, the existential $\langle\!\langle x \rangle\!\rangle$ and the universal $[\![x]\!]$, and an *agent binding* (a,x), where a is an agent and x is a variable. Intuitively, these new elements can be read, respectively, as *"there exists a strategy x"*, *"for all strategies x"*, and *"bind agent a to the strategy associated with variable x"*. The formal syntax of SL follows.

▶ **Definition 3.1** (Syntax)**.** SL *formulas* are built from the sets of atomic propositions AP, variables Var, and agents Ag, in the following way, where $p \in \mathrm{AP}$, $x \in \mathrm{Var}$, and $a \in \mathrm{Ag}$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\varphi \mid \varphi \mathsf{U} \varphi \mid \varphi \mathsf{R} \varphi \mid \langle\!\langle x \rangle\!\rangle \varphi \mid [\![x]\!]\varphi \mid (a,x)\varphi.$$

We now introduce some auxiliary syntactical notation for the definition of the semantics. By $\mathsf{free}(\varphi)$ we denote the set of *free agents/variables* of φ defined as the subset of $\mathrm{Ag} \cup \mathrm{Var}$ containing *(i)* all the agents for which there is no variable application after the occurrence of a temporal operator and *(ii)* all the variables for which there is an application but no quantifications. For example, let $\varphi = \langle\!\langle x \rangle\!\rangle (\alpha,x)(\beta,y)(\mathsf{F}\, p)$ be the formula on agents $\mathrm{Ag} = \{\alpha,\beta,\gamma\}$. Then, we have $\mathsf{free}(\varphi) = \{\gamma,y\}$, since γ is an agent without any application after $\mathsf{F}\, p$ and y has no quantification at all. A formula φ without free agents (resp., variables), i.e., with $\mathsf{free}(\varphi) \cap \mathrm{Ag} = \emptyset$ (resp., $\mathsf{free}(\varphi) \cap \mathrm{Var} = \emptyset$), is named *agent-closed* (resp., *variable-closed*). If φ is both agent- and variable-closed, it is named *sentence*.

Semantics. As for ATL*, we define the semantics of SL w.r.t. concurrent game structures. For a CGS $\mathcal{G}$, a state s, and an s-total assignment χ with $\mathsf{free}(\varphi) \subseteq \mathsf{dom}(\chi)$, we write $\mathcal{G},\chi,s \models \varphi$ to indicate that the formula φ holds at s under the assignment χ. Similarly, if χ is a complete assignment, for the (χ,s)-play π and a natural number k, we write $\mathcal{G},\chi,\pi,k \models \varphi$ to indicate that φ holds at the position k of π. The semantics of the SL formulas involving p, $\neg$, $\wedge$, and $\vee$, as well as that for the temporal operators X, U, and R, is defined as usual in LTL and we omit it here (see [13], for a survey). The semantics of the remaining part, which involves quantifications and bindings, follows.

▶ **Definition 3.2** (Semantics)**.** Given a CGS $\mathcal{G} = \langle \mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \lambda, \tau, s_0 \rangle$, an SL formula φ, a variable $x \in \mathrm{Var}$, a state $s \in \mathrm{St}$, and an s-total assignment χ with $\mathsf{free}(\varphi) \subseteq \mathsf{dom}(\chi) \cup \{x\}$, it holds that:

1. $\mathcal{G},\chi,s \models \langle\!\langle x \rangle\!\rangle \varphi$ iff there is an s-total strategy f such that $\mathcal{G},\chi[x \mapsto \mathsf{f}],s \models \varphi$;
2. $\mathcal{G},\chi,s \models [\![x]\!]\varphi$ iff for all s-total strategies f it holds that $\mathcal{G},\chi[x \mapsto \mathsf{f}],s \models \varphi$.

Moreover, if $\mathsf{free}(\varphi) \cup \{x\} \subseteq \mathsf{dom}(\chi) \cup \{a\}$ for an agent $a \in \mathrm{Ag}$, it holds that:

3. $\mathcal{G},\chi,s \models (a,x)\varphi$ iff $\mathcal{G},\chi[a \mapsto \chi(x)],s \models \varphi$.

Finally, if χ is also complete, where π is the (χ,s)-play and $k \in \mathbb{N}$, it holds that:

4. $\mathcal{G},\chi,s \models \varphi$ iff $\mathcal{G},\chi,\pi,0 \models \varphi$;
5. $\mathcal{G},\chi,\pi,k \models \varphi$ iff $\mathcal{G},\chi_{\pi_{\leq k}},\pi_k \models \varphi$.

Intuitively, at Items 1 and 2, respectively, we evaluate existential and universal quantifiers over strategies. At Item 3, by means of an agent binding (a,x), we commit the agent a to a strategy contained in the variable x. Finally, Items 4 and 5 can be easily understood by looking at their analogous path and state formulas in ATL*. In fact, Item 4 can be viewed as the rule that allows to move the verification process from states to paths and, vice versa, Item 5 from paths to states.

A CGS $\mathcal{G}$ is a *model* of an SL sentence φ, denoted by $\mathcal{G} \models \varphi$, iff $\mathcal{G},\varnothing,s_0 \models \varphi$, where $\varnothing$ is the empty assignment. Moreover, φ is *satisfiable* iff there is a model for it. For two SL formulas φ_1 and φ_2 we say that φ_1 is *equivalent* to φ_2, formally $\varphi_1 \equiv \varphi_2$, iff, for all CGSs $\mathcal{G}$, states s, and s-defined assignments χ with $\mathsf{free}(\varphi_1) \cup \mathsf{free}(\varphi_2) \subseteq \mathsf{dom}(\chi)$, it holds that $\mathcal{G},\chi,s \models \varphi_1$ iff $\mathcal{G},\chi,s \models \varphi_2$.

As an example, let $\varphi = \langle\!\langle x \rangle\!\rangle [\![y]\!] \langle\!\langle z \rangle\!\rangle (\alpha,x)(\beta,y)(\mathsf{X}\, p) \wedge (\alpha,y)(\beta,z)(\mathsf{X}\, q)$. First, note that α and β both use the strategy associated with y to achieve the goals $\mathsf{X}\, q$ and $\mathsf{X}\, p$, respectively. A model for φ is $\mathcal{G} = \langle \{p,q\}, \{\alpha,\beta\}, \{0,1\}, \{s_0,s_1,s_2,s_3\}, \lambda, \tau, s_0 \rangle$, where $\lambda(s_0) = \emptyset$, $\lambda(s_1) = \{p\}$, $\lambda(s_2) = \{p,q\}$, $\lambda(s_3) = \{q\}$, $\tau(s_0,(0,0)) = s_1$, $\tau(s_0,(0,1)) = s_2$, $\tau(s_0,(1,0)) = s_3$, and all the remaining transitions (with any action) go to s_0. Clearly, $\mathcal{G},s_0 \models \varphi$ by letting, on s_0, x to chose action 0 (the goal $\mathsf{X}\, p$ is

satisfied for any choice of y, since we can move from s_0 to s_1 or s_2, both labeled with p) and z to choose action 1 when y has action 0 and, vice versa, z to 0 when y has 1 (in both the cases, the goal $\mathsf{X}\,q$ is satisfied, since one can move from s_0 to s_2 or s_3, both labeled with q).

An important property that is possible to express in SL, but neither in ATL* nor in $\mathrm{SL}_{\mathrm{CHP}}$, is the existence of *deterministic multi-player Nash equilibria*. For example, consider n agents $\alpha_1,\ldots,\alpha_n$ each of them having the LTL goals $\psi_1,\ldots,\psi_n$. Then, we can express the existence of a *strategy profile* $(x_1,\ldots,x_n)$ that is a Nash equilibrium for $\alpha_1,\ldots,\alpha_n$ w.r.t. $\psi_1,\ldots,\psi_n$ by using the sentence $\langle\!\langle x_1\rangle\!\rangle\cdots\langle\!\langle x_n\rangle\!\rangle(\alpha_1,x_1)\cdots(\alpha_n,x_n)(\bigwedge_{i=1}^{n}(\langle\!\langle y\rangle\!\rangle(\alpha_i,y)\psi_i)\rightarrow\psi_i)$. Informally, this sentence asserts that every agent has the "best" strategy once all the strategies of the remaining agents have been fixed. Note that here we have only considered equilibria under deterministic strategies.

Basic properties. We now investigate some basic properties of SL that turn out to be important for their own and useful to prove the decidability of the model checking problem and the undecidability of the satisfiability one. In particular, for the introduced logic we investigate the tree and finite model properties. To this aim, we define a generalization to CGS of the classical concept of unwinding of labeled transition systems, which allows us to show that SL has the (general) tree model property, but neither the bounded-tree model property nor the finite model property. As preliminary, we need to formally state the concepts of *concurrent game tree* and *unwinding* of a game structure.

A *concurrent game tree* (CGT, for short) is a CGS $\mathcal{U}=\langle \mathrm{AP},\mathrm{Ag},\mathrm{Ac},\mathrm{St},\lambda,\tau,\varepsilon\rangle$, where $\mathrm{St}\subseteq\Delta^*$ is a tree for a given set of directions Δ and $s\cdot t\in\mathrm{St}$ iff there is a decision $\mathsf{d}\in\mathrm{Dc}$ such that $\tau(s,\mathsf{d})=s\cdot t$, for all $s\in\mathrm{St}$ and $t\in\Delta$. For a CGS $\mathcal{G}=\langle \mathrm{AP},\mathrm{Ag},\mathrm{Ac},\mathrm{St},\lambda,\tau,s_0\rangle$, the *unwinding* of $\mathcal{G}$ is the CGT $\mathcal{G}^{\mathcal{U}}\triangleq\langle \mathrm{AP},\mathrm{Ag},\mathrm{Ac},\mathrm{St}',\lambda',\tau',\varepsilon\rangle$, where St is the set of directions of the tree, the states in $\mathrm{St}'=\{\rho\in\mathrm{St}^* \mid s_0\cdot\rho\in\mathrm{Trk}\}$ are the suffixes of the tracks starting in s_0, and, for all $s\in\mathrm{St}'$, $\mathsf{d}\in\mathrm{Dc}$, and $t\in\mathrm{St}$, it holds that $\tau'(s,\mathsf{d})=s\cdot\tau(\mathsf{lst}(s),\mathsf{d})$ and there is a surjective function $\mathsf{unw}:\mathrm{St}'\rightarrow\mathrm{St}$ such that *(i)* $\mathsf{unw}(\varepsilon)=s_0$, *(ii)* $\mathsf{unw}(s\cdot t)=t$, and *(iii)* $\lambda'(s)=\lambda(\mathsf{unw}(s))$.

We say that SL has the *tree model property* if every satisfiable formula is satisfiable by a CGT. Actually, next theorem reports that SL is invariant under unwinding, so, it has the tree model property. This can be shown by using a proof by induction on the structure of the formula, making use of the unwinding technique defined above.

▶ **Theorem 3.3.** *$\mathcal{G}\models\varphi$ iff $\mathcal{G}^{\mathcal{U}}\models\varphi$.*

We now move to the negative results about SL, namely, that it neither has the finite model property nor the bounded-tree model property. We recall that a modal logic has the bounded-tree model property (resp., finite model property) if whenever a formula is satisfiable, it is so on a model having a tree shape (resp., a finite number of states) in which every state has at most n successors, for a natural number n. Clearly, if a modal logic with the tree model property has the finite model property, it has the bounded-tree model property as well. The other direction may not hold, instead. To prove both results, we introduce in Definition 3.4 the formula φ^{ord} to be used as a counterexample.

▶ **Definition 3.4.** Let $x_1<x_2\triangleq\langle\!\langle y\rangle\!\rangle\,\varphi^{in}(x_1,x_2,y)$, where $\varphi^{in}(x_1,x_2,y)\triangleq(\beta,y)((\alpha,x_1)(\mathsf{X}\,p)\wedge(\alpha,x_2)(\mathsf{X}\neg p))$ is an agent-closed SL formula, named *partial order*, on the sets $\mathrm{AP}=\{p\}$ and $\mathrm{Ag}=\{\alpha,\beta\}$. Then, the SL *order sentence* $\varphi^{ord}\triangleq\varphi^{unb}\wedge\varphi^{trn}$ is the conjunction of the following two sentences, called *strategy unboundedness* and *strategy transitivity* requirements:

1. $\varphi^{unb}\triangleq[\![x_1]\!]\langle\!\langle x_2\rangle\!\rangle\, x_1<x_2$;
2. $\varphi^{trn}\triangleq[\![x_1]\!][\![x_2]\!][\![x_3]\!]\,(x_1<x_2\wedge x_2<x_3)\rightarrow x_1<x_3$.

Intuitively, φ^{unb} asserts that, for each strategy x_1, there is a different strategy x_2 in relation of $<$ w.r.t. the first one, i.e., $<$ has no upper bound, while φ^{trn} expresses the fact that the relation $<$ is transitive. Note also that, by definition, $<$ is not reflexive. Obviously, the formula φ^{ord} needs to be satisfiable, as reported in the following lemma.

▶ **Lemma 3.5.** *The* SL *sentence* φ^{ord} *is satisfiable.*

Next two lemmas report two important properties of the formula φ^{ord}, for the negative statements we want to show. Namely, they state that, in order to be satisfied, φ^{ord} must require the existence of strict partial order relations on strategies and actions that do not admit any maximal element. From this, as stated in Theorem 3.8, we directly derive that φ^{ord} needs an infinite chain of actions to be satisfied (i.e., it cannot have a bounded model).

▶ **Lemma 3.6.** *Let* $\mathcal{G}$ *be a model of* φ^{ord} *and* $r^{<} \subseteq \mathrm{Str} \times \mathrm{Str}$ *be a relation between strategies such that* $r^{<}(\mathsf{f}_1, \mathsf{f}_2)$ *holds iff* $\mathcal{G}, \chi, s_0 \models x_1 < x_2$*, where* $\chi(x_1) = \mathsf{f}_1$ *and* $\chi(x_2) = \mathsf{f}_2$*, for all* $\chi \in \mathrm{Asg}$*, with* s_0 *as the initial state of* $\mathcal{G}$*. Then* $r^{<}$ *is a* strict partial order without maximal element.

▶ **Lemma 3.7.** *Let* $\mathcal{G}$ *be a model of* φ^{ord} *and* $s^{<} \subseteq \mathrm{Ac} \times \mathrm{Ac}$ *be a relation between actions such that* $s^{<}(c_1, c_2)$ *holds iff* $r^{<}(\mathsf{f}_1, \mathsf{f}_2)$ *holds, where* $c_1 = \mathsf{f}_1(s_0)$ *and* $c_2 = \mathsf{f}_2(s_0)$*, for all* $\mathsf{f}_1, \mathsf{f}_2 \in \mathrm{Str}$*, with* s_0 *as the initial state of* $\mathcal{G}$*. Then* $s^{<}$ *is a* strict partial order without maximal element.

Observe that the relation $s^{<}$ cannot be defined on a finite set [6]. Now, we have all tools to prove that SL lacks of the finite and bounded-tree model properties, which hold in several commonly used multi-agent logics, such as ATL*.

▶ **Theorem 3.8.** *For* SL *it holds that:* (i) *it does not have the bounded-tree model property, and* (ii) *it does not have the finite model property.*

4 Model Checking

In this section, we study the model-checking problem for SL and show that it is decidable and 2ExpTime-Complete, as for ATL*. The lower bound immediately follows from ATL*, which SL properly includes. For the upper bound, we follow an *automata-theoretic approach* [13], reducing the decision problem for the logic of interest to the emptiness problem of automata. To this aim, we use *alternating parity tree automata* (APT, for short), which are *alternating tree automata* along with a *parity acceptance condition* (see [8], for a survey). APTs are a generalization of *nondeterministic parity tree automata* (NPT, for short). Intuitively, while an NPT that visits a node of the input tree sends exactly one copy of itself to each of the successors of the node, an APT can send several copies of itself to the same successor. We recall that an approach to tree automata is only possible once the logic satisfies the tree model property. In fact, this property holds for SL as we have proved in Theorem 3.3. By the size of the automaton and the complexity required for checking its emptiness, we get the desired 2ExpTime upper bound. The definition of APTs follows.

▶ **Definition 4.1.** An APT is a tuple $\mathcal{A} = \langle \Sigma, \Delta, \mathrm{Q}, \delta, q_0, \mathrm{F} \rangle$, where Σ, Δ, and Q are non-empty finite sets of *input symbols*, *directions*, and *states*, $q_0 \in \mathrm{Q}$ is an *initial state*, $\mathrm{F} = (\mathrm{F}_1, \ldots, \mathrm{F}_k) \in (2^{\mathrm{Q}})^*$ with $\mathrm{F}_1 \subseteq \ldots \subseteq \mathrm{F}_k = \mathrm{Q}$ is a *parity acceptance condition*, and $\delta : \mathrm{Q} \times \Sigma \to \mathtt{B}^+(\Delta \times \mathrm{Q})$ is an *alternating transition function* that maps each pair of states and symbols into a Boolean positive formula on the set of propositions of the form (d, q), where d is a direction and q a state.

A *run* of an APT $\mathcal{A}$ on a Σ-labeled Δ-tree $\mathcal{T} = \langle \mathrm{T}, \mathsf{v} \rangle$ is a $(\mathrm{Q} \times \mathrm{T})$-labeled $\mathbb{N}$-tree $\mathcal{R} = \langle \mathrm{Tr}, \mathsf{r} \rangle$ such that *(i)* $\mathsf{r}(\varepsilon) = (q_0, \varepsilon)$ and *(ii)* for all $y \in \mathrm{Tr}$ with $\mathsf{r}(y) = (q, x)$, there exists a set $S \subseteq \mathrm{D} \times \mathrm{Q}$ with $S \models \delta(q, \mathsf{v}(x))$ such that, for all atoms $(d, q') \in S$, there is an index $i \in \mathbb{N}$ for which it holds that $\mathsf{r}(y \cdot i) = (q', x \cdot d)$. The run $\mathcal{R}$ is said to be *accepting* iff, for every path π, the least index $1 \leq i \leq k$ such that at least one state of F_i occurs infinitely often in π is even. The number k of sets in F is called the *index* of the automaton. A tree $\mathcal{T}$ is *accepted* by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on it. By $\mathrm{L}(\mathcal{A})$ we denote the language accepted by the automaton $\mathcal{A}$, i.e., the set of all trees that $\mathcal{A}$ accepts. $\mathcal{A}$ is said *empty* if $\mathrm{L}(\mathcal{A}) = \emptyset$. The *emptiness problem* for $\mathcal{A}$ is to decide whether $\mathrm{L}(\mathcal{A}) = \emptyset$.

We now proceed with the model-checking algorithm for SL. As for ATL*, we use a bottom-up model-checking algorithm, in which we start with the innermost sub-sentences and terminate with the sentence under checking. At each step, we label each state of the model with all the sub-sentences that are satisfied on it. The procedure we propose here extends that used for ATL* in [1] by means of a richer structure of the automata involved in.

First, we introduce some extra notation. A *principal sentence* φ is a sentence of the form $\mathsf{Qn}_1 x_1 \cdots \mathsf{Qn}_k x_k\, \psi_\varphi$, where $\mathsf{Qn}_i x_i \in \{\langle\!\langle x_i \rangle\!\rangle, [\![x_i]\!]\}$ and the *matrix* ψ_φ is an agent-closed formula, with $\mathsf{free}(\psi_\varphi) = \{x_1, \ldots, x_k\}$, such that it does not contain any quantification. For the sake of space and clarity of exposition, we only discuss the model checking of principal formulas. By a slight variation of both the notion of principal formulas and our procedure, we can also address the full SL. We also need the notion of atom. An *atom* ψ is an agent-closed formula of the form $(\alpha_1, y_1) \cdots (\alpha_n, y_n)\psi'$, where $\mathrm{Ag} = \{\alpha_1, \ldots, \alpha_n\}$, $y_1, \ldots, y_n$ are possible equal variables and either *(i)* ψ' does not contain any quantification and binding, i.e., it is an LTL formula, or *(ii)* the derived formula $\widehat{\psi}'$ does not contain any quantification and binding at all, where $\widehat{\psi}'$ is obtained by ψ' substituting its sub-atoms with fresh atomic propositions. W.l.o.g., we assume that each principal sentence has a matrix that is a Boolean combination of atoms. $\mathrm{Atm}(\varphi)$ denotes the set of all sub-formulas of φ that are atoms.

The core idea behind our model-checking procedure is the following. Let $\mathcal{G} = \langle \mathrm{AP}, \mathrm{Ag}, \mathrm{Ac}, \mathrm{St}, \lambda, \tau, s_0 \rangle$ be a CGS and φ be an SL principal sentence over the set $\mathrm{Ag} = \{\alpha_1, \ldots, \alpha_n\}$ of n different agents, for which we want to check if $\mathcal{G} \models \varphi$ holds or not. We first build an NPT $\mathcal{D}_\mathcal{G}$ recognizing the unwinding $\mathcal{G}^\mathcal{U}$ of $\mathcal{G}$. Then, we build an APT $\mathcal{A}'_{\mathcal{G},\varphi}$ accepting all prunings of $\mathcal{G}^\mathcal{U}$ that are coherent with the strategy quantification of φ. Such prunings are done by properly labeling its paths with elements from the set $\mathrm{Z} \triangleq \mathrm{Atm}(\varphi) \times \{start, pass\}$ of atoms associated with a flag in $\{start, pass\}$, in a way similar as it has been done for ATL* satisfiability in [20]. The *start* and *pass* flags are used to indicate whether a path guessed to satisfy at a specific state an atom $\psi \in \mathrm{Atm}(\varphi)$, starts or passes through that state, respectively. Namely, the unlabeled paths are the pruned ones that are not needed in order to satisfy the formula. Hence, $\mathcal{A}'_{\mathcal{G},\varphi}$ accepts $\mathcal{G}^\mathcal{U}$ with this additional labeling. The automata $\mathcal{D}_\mathcal{G}$ and $\mathcal{A}'_{\mathcal{G},\varphi}$ have index 2 and a number of states polynomial in the size of $\mathcal{G}$ and φ, respectively. With more details, they are both safety automata[1]. Finally, we build an APT $\mathcal{A}''_\varphi$ that checks that all paths of a pruned model accepted by $\mathcal{A}'_{\mathcal{G},\varphi}$, i.e., all labeled paths, satisfy the atoms of φ. The automaton $\mathcal{A}''_\varphi$ has index 2 and a number of states exponential in φ.

Now, recall that APTs are linearly closed under intersection. More precisely, two APTs having n_1 and n_2 states and k_1 and k_2 as indexes, respectively, can be intersected in an APT with $n_1 + n_2$ states and index $\max\{k_1, k_2\}$ [15]. So, we can build an APT $\mathcal{A}_{\mathcal{G},\varphi}$ such that $\mathrm{L}(\mathcal{A}_{\mathcal{G},\varphi}) = \mathrm{L}(\mathcal{A}'_{\mathcal{G},\varphi}) \cap \mathrm{L}(\mathcal{A}''_\varphi)$, having in particular index 2. Also, by [15], we can translate an APT with n states and index k in an equivalent NPT having $n^{\mathtt{O}(n)}$ states and index $\mathtt{O}(n)$. Hence, we can transform $\mathcal{A}_{\mathcal{G},\varphi}$ in an NPT $\mathcal{N}_{\mathcal{G},\varphi}$ with a number of states double exponential in φ and an index exponential in φ. It is well known that an NPT having n states and index k and a safety automaton with m states can be intersected in an NPT with $n \cdot m$ states and index k. Hence, by intersecting $\mathcal{D}_\mathcal{G}$ with $\mathcal{N}_{\mathcal{G},\varphi}$, we get an NPT $\mathcal{N}'_{\mathcal{G},\varphi}$ such that $\mathrm{L}(\mathcal{N}'_{\mathcal{G},\varphi}) = \mathrm{L}(\mathcal{D}_\mathcal{G}) \cap \mathrm{L}(\mathcal{N}_{\mathcal{G},\varphi})$. At this point, it is possible to prove that $\mathcal{G} \models \varphi$ iff $\mathrm{L}(\mathcal{N}'_{\mathcal{G},\varphi}) \neq \emptyset$. Observe that $\mathcal{N}'_{\mathcal{G},\varphi}$ has a number of states double exponential in φ and polynomial in $\mathcal{G}$, while it has an index exponential in φ, but independent from $\mathcal{G}$. Moreover, the automata run over the alphabet $\Sigma = \{\sigma \subseteq \mathrm{AP} \cup \mathrm{St} \cup \mathrm{Z} \mid |\sigma \cap \mathrm{St}| = 1\}$, where $|\mathrm{Z}| = \mathtt{O}(|\mathcal{G}| \times 2^{|\varphi|})$. Since the emptiness of an NPT with n states, index k, and alphabet size h can be checked in time $\mathtt{O}(h \cdot n^k)$ [12], we get that to check whether $\mathcal{G} \models \varphi$ can be done in time double exponential in φ and polynomial in $\mathcal{G}$. More precisely, the algorithm runs in $|\mathcal{G}|^{2^{\mathtt{O}(|\varphi|)}}$. The details of the automata construction follow.

[1] A safety condition is the special parity condition $(\emptyset, \mathrm{Q})$ of index 2.

The NPT $\mathcal{D}_\mathcal{G} = \langle \Sigma, \mathrm{St}, \mathrm{St}, \delta, s_0, (\emptyset, \mathrm{St}) \rangle$ has the set of directions and states formed by the states of $\mathcal{G}$ that are used to build its unwinding. Moreover, the transition function is defined as follows. At the state $s \in \mathrm{St}$, the automaton first checks that the labeling of the node of the input tree corresponds to the union of $\{s\}$ and its labeling $\lambda(s)$ in $\mathcal{G}$. Then, it sends all successors of s in the relative directions. Formally, $\delta(s,\sigma)$ is set to $\mathfrak{f}$ (false) if $\lambda(s) \cup \{s\} \neq \sigma \cap (\mathrm{AP} \cup \mathrm{St})$ and to $\bigwedge_{s' \in \{\tau(s,\mathsf{d}) \mid \mathsf{d} \in \mathrm{Dc}\}} (s', s')$ otherwise. Note that $|\mathcal{D}_\mathcal{G}| = \mathtt{O}(|\mathcal{G}|)$.

The APT $\mathcal{A}'_{\mathcal{G},\varphi} = \langle \Sigma, \mathrm{St}, \{q_0\} \cup \mathrm{Atm}(\varphi), \delta, q_0, (\emptyset, \{q_0\} \cup \mathrm{Atm}(\varphi)) \rangle$ has the set of states formed by a distinguished state q_0, which is also initial, and from the atoms in $\mathrm{Atm}(\varphi)$ that are used to verify the correctness of the additional labeling Z. Moreover, the transition function is defined as follows. $\delta(\psi,\sigma)$ is equal to $\mathfrak{t}$ (true) if $(\psi, \mathit{pass}) \in \sigma \cap \mathrm{Z}$ and to $\mathfrak{f}$ (false) otherwise. The automaton at state q_0 sends the same state in all the directions individuated by the quantification, together with the control state ψ. It is important to note that the quantification here is reproduced by conjunctions and disjunctions on all possible actions of $\mathcal{G}$. Formally, $\delta(q_0,\sigma)$ is set to $\mathsf{Op}_{1\,c_1 \in \mathrm{Ac}} \cdots \mathsf{Op}_{k\,c_k \in \mathrm{Ac}} \bigwedge_{(\psi,\star) \in \sigma \cap \mathrm{Z}} (\tau(s,\mathsf{d}), q_0) \wedge (\tau(s,\mathsf{d}), \psi)$, where $\mathsf{Op}_{i\,c_i \in \mathrm{Ac}}$ is a disjunction if $\mathsf{Qn}_i x_i = \langle\!\langle x_i \rangle\!\rangle$ and a conjunction if $\mathsf{Qn}_i x_i = [\![x_i]\!]$, $\{s\} = \sigma \cap \mathrm{St}$, and $\mathsf{d}(\alpha_i) = c_j$ iff in the atom ψ the binding (α_i, x_j) appears. Note that $|\mathcal{A}'_{\mathcal{G},\varphi}| = \mathtt{O}(|\varphi|)$.

Finally, we build the APT $\mathcal{A}''_\varphi$. Let $\widehat{\psi}$ be the LTL formula obtained by replacing in $\psi \in \mathrm{Atm}(\varphi)$ all the occurrences of each other atom $\psi' \in \mathrm{Atm}(\psi)$ with the fresh atomic proposition (ψ', start). By using a slight variation of the procedure developed in [21], we can translate $\widehat{\psi}$ into a universal co-Büchi word automaton[2] $\mathcal{U}_\psi = \langle \Sigma, \mathrm{Q}_\psi, \delta_\psi, \mathrm{Q}_{0\psi}, \mathrm{F}_\psi \rangle$, with a number of states at most exponential in $|\psi|$, accepting the infinite words on Σ that are models of $\widehat{\psi}$. At this point, we can construct the automaton $\mathcal{A}''_\varphi$ that recognizes the trees whose paths, labeled with the flags $(\psi, \star)$, for $\star \in \{\mathit{start}, \mathit{pass}\}$, and starting with the label (ψ, start), satisfy the LTL formula $\widehat{\psi}$, for all $\psi \in \mathrm{Atm}(\varphi)$.

Formally, $\mathcal{A}''_\varphi = \langle \Sigma, \mathrm{St}, \{q_0, q_c\} \cup \mathrm{Q}, \delta, q_0, (\mathrm{F}, \{q_0, q_c\} \cup \mathrm{Q}) \rangle$ is built as follows. $\mathrm{Q} = \bigcup_{\psi \in \mathrm{Atm}(\varphi)} \{\psi\} \times \mathrm{Q}_\psi$ and $\mathrm{F} = \bigcup_{\psi \in \mathrm{Atm}(\varphi)} \{\psi\} \times \mathrm{F}_\psi$ are, respectively, the disjoint union of the set of states and final states of the word automata $\mathcal{U}_\psi$, for every atom $\psi \in \mathrm{Atm}(\varphi)$. q_0 is the *initial state* used to verify that the formula ψ_φ (the matrix of φ) holds at the root of the tree in input, by checking whether the labeling of the root contains all the propositions required by ψ_φ to hold. If the checking succeeds, q_0 behaves as the state q_c. Formally, let ψ_φ be considered as a boolean formula on the set of atoms $\mathrm{Atm}(\varphi)$ in which we assume $\psi = (\psi, \mathit{start})$, for all $\psi \in \mathrm{Atm}(\varphi)$. Then, $\delta(q_0,\sigma)$ is set to $\delta(q_c,\sigma)$, if $\sigma \cap \mathrm{Z} \models \psi_\varphi$ and to $\mathfrak{f}$ (false), otherwise. q_c is the *checking state* used to start the verification of the atoms ψ in every node of the input tree that contains the flag (ψ, start), which indicates the existence of a path starting in that node that satisfies ψ. To do this, q_c sends in all the directions *(i)* a copy of the state itself, to continue the control on the remaining part of the tree, and *(ii)* the states derived by all initial states of the automata $\mathcal{U}_\psi$, for all the atoms ψ for which a flag (ψ, start) appears in the labeling σ. Formally, $\delta(q_c,\sigma)$ is $\bigwedge_{s \in \mathrm{St}} (s, q_c) \wedge \bigwedge_{(\psi,\mathit{start}) \in \sigma \cap \mathrm{Z}} \bigwedge_{q \in \mathrm{Q}_{0\psi}} \bigwedge_{q' \in \delta_\psi(q, \sigma \cap \mathrm{AP})} (s, (\psi, q'))$. The states of the form (ψ, q) are used to run $\mathcal{U}_\psi$ on all paths labeled by the related flags (ψ, pass). Formally, $\delta((\psi, q), \sigma)$ is set to $\mathfrak{t}$ (true) if $(\psi, \mathit{pass}) \notin \sigma \cap \mathrm{Z}$ and to $\bigwedge_{s \in \mathrm{St}} \bigwedge_{q' \in \delta_\psi(q, \sigma \cap \mathrm{AP})} (s, (\psi, q'))$ otherwise. Note that $|\mathcal{A}''_\varphi| = \mathtt{O}(2^{|\varphi|})$.

By a simple calculation, it follows that the overall procedure results in an algorithm that is in PTIME w.r.t the size of $\mathcal{G}$ and in 2EXPTIME w.r.t. the size of φ. Hence, by getting lower bounds from ATL*, the following result holds.

▶ **Theorem 4.2.** *The model-checking problem for* SL *is* PTIME *w.r.t. the size of the model and* 2EXPTIME-COMPLETE *w.r.t the size of the specification.*

We conclude this section by pointing out that the model checking procedure described above for SL is completely different from that one used in [3] for $\mathrm{SL}_{\mathrm{CHP}}$. Indeed in [3], the authors use

[2] Word automata can be seen as tree automata in which the tree has just one path. A universal word automaton is a particular case of alternating automata in which there is no nondeterminism. A co-Büchi acceptance condition $\mathrm{F} \subseteq \mathrm{Q}$ is the special parity condition (F, Q) of index 2.

a top-down approach and, most important, for every quantification in the formula, they make a projection of the automaton they build at each stage (one for each quantification). Since at each projection they have an exponential blow-up, at the end their procedure results in a non-elementary one, both in the size of the system and the formula. Our iterative approach, instead, does not make use of any projection, since we reduce strategy quantifications to action quantifications, which, as we have stated, can be handled locally on each state of the model.

5 Satisfiability

In this section, we show the undecidability of the satisfiability problem for SL through a reduction of the *recurrent domino problem*. In particular, as we discuss later, the reduction also holds for SL_{CHP} under the concurrent game semantics.

The *domino problem*, proposed for the first time by Wang [22], consists of placing a given number of tile types on an infinite grid, satisfying a predetermined set of constraints on adjacent tiles. Its standard version asks for a compatible tiling of the whole plane $\mathbb{N}\times\mathbb{N}$. The *recurrent domino problem* further requires the existence of a distinguished tile type that occurs infinitely often in the first row of the grid. This problem was proved to be highly undecidable by Harel, and in particular, Σ^1_1-COMPLETE [9]. The formal definition follows.

▶ **Definition 5.1** (Recurrent Domino System)**.** An $\mathbb{N}\times\mathbb{N}$ *recurrent domino system* $\mathcal{D}=\langle \mathrm{D}, H, V, t^*\rangle$ consists of a finite non-empty set D of *domino types*, two *horizontal* and *vertical matching relations* $H, V \subseteq \mathrm{D}\times\mathrm{D}$, and a distinguished tile type $t^*\in\mathrm{D}$. The recurrent domino problem asks for an *admissible tiling* of $\mathbb{N}\times\mathbb{N}$, which is a *solution mapping* $\partial:\mathbb{N}\times\mathbb{N}\to\mathrm{D}$ such that, for all $x,y\in\mathbb{N}$, it holds that *(i)* $(\partial(x,y),\partial(x+1,y))\in H$, *(ii)* $(\partial(x,y),\partial(x,y+1))\in V$, and *(iii)* $|\{x \mid \partial(x,0)=t^*\}|=\infty$.

By showing a reduction from the recurrent domino problem, we prove that the satisfiability problem for SL is Σ^1_1-HARD, which implies that it is even not computably enumerable. We achieve this reduction by showing that a given recurrent tiling system $\mathcal{D}=\langle \mathrm{D}, H, V, t^*\rangle$ can be *"embedded"* into a model of a particular sentence $\varphi^{dom} \triangleq \varphi^{grd}\wedge\varphi^{til}\wedge\varphi^{rec}$ over $\mathrm{AP}=\{p\}\cup\mathrm{D}$ and $\mathrm{Ag}=\{\alpha,\beta\}$, where $p\notin\mathrm{D}$, in such a way that φ^{dom} is satisfiable iff $\mathcal{D}$ allows an admissible tiling. For the sake of clarity, we split the reduction into three tasks where we explicit the sentences φ^{grd}, φ^{til}, and φ^{rec}.

Grid specification. Consider the sentence $\varphi^{grd}\triangleq\bigwedge_{a\in\mathrm{Ag}}\varphi^{ord}_a$, where $\varphi^{ord}_a=\varphi^{unb}_a\wedge\varphi^{trn}_a$ are *order sentences* and φ^{exs}_a and φ^{trn}_a are the *strategy unboundedness* and *strategy transitivity* requirements for agents α and β defined, similarly in Definition 3.4, as follows:

1. $\varphi^{unb}_a\triangleq[\![z_1]\!]\langle\!\langle z_2\rangle\!\rangle\, z_1<_a z_2$,
2. $\varphi^{trn}_a\triangleq[\![z_1]\!][\![z_2]\!][\![z_3]\!]\,(z_1<_a z_2\wedge z_2<_a z_3)\to z_1<_a z_3$,

where $x_1<_\alpha x_2\triangleq\langle\!\langle y\rangle\!\rangle\,(\beta,y)((\alpha,x_1)(\mathsf{X}\,p)\wedge(\alpha,x_2)(\mathsf{X}\neg p))$ and $y_1<_\beta y_2\triangleq\langle\!\langle x\rangle\!\rangle\,(\alpha,x)((\beta,y_1)(\mathsf{X}\neg p)\wedge(\beta,y_2)(\mathsf{X}\,p))$ are the two *partial order* formulas on strategies of α and β, respectively. Intuitively, $<_\alpha$ and $<_\beta$ correspond to the horizontal and vertical ordering of the positions in the grid, respectively.

It easy to see that φ^{grd} is satisfiable, as it follows from the same argument used in Lemma 3.5.

▶ **Lemma 5.2.** *The* SL *sentence* φ^{grd} *is satisfiable on a* CGS *with a countable number of actions.*

Consider now a model $\mathcal{G}=\langle \mathrm{AP},\mathrm{Ag},\mathrm{Ac},\mathrm{St},\lambda,\tau,s_0\rangle$ of φ^{grd} and the relations $r^<_a\subseteq\mathrm{Str}\times\mathrm{Str}$ between strategies, for all agents $a\in\mathrm{Ag}$, defined as follows: $r^<_a(\mathsf{f}_1,\mathsf{f}_2)$ holds iff $\mathcal{G},\chi,s_0\models z_1<_a z_2$, where $\chi(z_1)=\mathsf{f}_1$ and $\chi(z_2)=\mathsf{f}_2$. By Lemma 3.6, the relations $r^<_a$ are *strict partial orders without maximal element* on Str. To apply the desired reduction, we need to transform $r^<_a$ into total orders over strategies. Let $r^{\equiv}_a\subseteq\mathrm{Str}\times\mathrm{Str}$, with $a\in\mathrm{Ag}$, be the two relations between strategies such that $r^{\equiv}_a(\mathsf{f}_1,\mathsf{f}_2)$ holds iff neither $r^<_a(\mathsf{f}_1,\mathsf{f}_2)$ nor $r^<_a(\mathsf{f}_2,\mathsf{f}_1)$ holds. It is possible to show that $r^{\equiv}_a$ are *equivalence relations*. Now, let $\mathrm{Str}^{\equiv}_a=\mathrm{Str}/r^{\equiv}_a$ be the quotient sets of Str w.r.t. $r^{\equiv}_a$, i.e., the sets of the related equivalence

classes over strategies. Also, let $s_a^{<} \subseteq \mathrm{Str}_a^{\equiv} \times \mathrm{Str}_a^{\equiv}$, with $a \in \mathrm{Ag}$, be the two relations between classes of strategies such that $s_a^{<}(F_1, F_2)$ holds iff, for all $\mathsf{f}_1 \in F_1$ and $\mathsf{f}_2 \in F_2$, it holds that $r_a^{<}(\mathsf{f}_1, \mathsf{f}_2)$. Then, it is easy to prove that $s_a^{<}$ are *strict total orders with minimal element but no maximal element*. By a classical result on first order logic model theory [6], $s_a^{<}$ cannot be defined on a finite set. Hence, $|\mathrm{Str}_a^{\equiv}| = \infty$, for all $a \in \mathrm{Ag}$. Now, let $s_a^{\prec}$ be the *successor* relations on $\mathrm{Str}_a^{\equiv}$ compatible with the strict total orders $s_a^{<}$, i.e., such that $s_a^{\prec}(F_1, F_2)$ holds iff *(i)* $s_a^{<}(F_1, F_2)$ holds and *(ii)* there is no $F_3 \in \mathrm{Str}_a^{\equiv}$ for which both $s_a^{<}(F_1, F_3)$ and $s_a^{<}(F_3, F_2)$ hold, for all $F_1, F_2 \in \mathrm{Str}_a^{\equiv}$. Then, we can write the two sets of classes $\mathrm{Str}_\alpha^{\equiv}$ and $\mathrm{Str}_\beta^{\equiv}$ as the infinite ordered lists $\{F_0^\alpha, F_1^\alpha, \ldots\}$ and $\{F_0^\beta, F_1^\beta, \ldots\}$, respectively, such that $s_a^{\prec}(F_i^a, F_{i+1}^a)$, for all $a \in \mathrm{Ag}$ and $i \in \mathbb{N}$. Note that F_0^a are the classes of minimal strategies w.r.t the relations $s_a^{<}$.

Now, we have all the machinery to build an embedding of the plane $\mathbb{N} \times \mathbb{N}$ into the model $\mathcal{G}$ of φ^{grd}. In particular, we are able to construct a *bijective map* $\aleph : \mathbb{N} \times \mathbb{N} \to \mathrm{Str}_\alpha^{\equiv} \times \mathrm{Str}_\beta^{\equiv}$ such that $\aleph(i, j) = (F_i^\alpha, F_j^\beta)$, for all $i, j \in \mathbb{N}$.

Compatible tiling. Given the grid structure built on the model $\mathcal{G}$ of φ^{grd} through the bijective map $\aleph$, we can express that a tiling of the grid is admissible by making use of the formulas $z_1 \prec_a z_2 \triangleq z_1 <_a z_2 \wedge \neg\langle\!\langle z_3 \rangle\!\rangle\, z_1 <_a z_3 \wedge z_3 <_a z_2$ corresponding to the successor relations $s_a^{\prec}$, for all $a \in \mathrm{Ag}$. Indeed, it is not hard to see that $\mathcal{G}, \chi, \varepsilon \models z_1 \prec_a z_2$ holds iff $\chi(z_1) \in F_i^a$ and $\chi(z_2) \in F_{i+1}^a$, for all $i \in \mathbb{N}$. The idea here is to associate to each domino type $t \in \mathrm{D}$ a corresponding atomic proposition $t \in \mathrm{AP}$ and to express the horizontal and vertical matching conditions via suitable object labeling. In particular, we can express that the tiling is locally compatible, that the horizontal neighborhood of a tile satisfies the H requirement, and that also its vertical neighborhood satisfies the V requirement, all through the following three agent-closed formulas, respectively:

1. $\varphi^{t,loc}(x,y) \triangleq (\alpha,x)(\beta,y)(\mathsf{X}\,(t \wedge \bigwedge_{t' \in \mathrm{D}}^{t' \neq t} \neg t'))$;
2. $\varphi^{t,hor}(x,y) \triangleq \bigvee_{(t,t') \in H} [\![x']\!]\, x \prec_\alpha x' \to (\alpha,x')(\beta,y)(\mathsf{X}t')$;
3. $\varphi^{t,ver}(x,y) \triangleq \bigvee_{(t,t') \in V} [\![y']\!]\, y \prec_\beta y' \to (\alpha,x)(\beta,y')(\mathsf{X}t')$.

Informally, we have the following: $\varphi^{t,loc}(x,y)$ asserts that t is the only domino type labeling the successors of the root of the model $\mathcal{G}$ that can be reached using the strategies related to the variables x and y; $\varphi^{t,hor}(x,y)$ asserts that the tile t' labeling the successors of the root reachable through the strategies x' and y is compatible with t w.r.t. the horizontal requirement H, for all strategies x' that immediately follow that related to x w.r.t. the order $r_\alpha^{<}$; $\varphi^{t,ver}(x,y)$ asserts that the tile t' labeling the successors of the root reachable through the strategies x and y' is compatible with t w.r.t. the vertical requirement V, for all strategies y' that immediately follow that related to y w.r.t. the order $r_\beta^{<}$.

Finally, to express that the whole grid has an admissible tiling, we use the sentence $\varphi^{til} \triangleq [\![x]\!][\![y]\!] \bigvee_{t \in \mathrm{D}} \varphi^{t,loc}(x,y) \wedge \varphi^{t,hor}(x,y) \wedge \varphi^{t,ver}(x,y)$ that asserts, for every point individuated by the strategies x and y, the existence of a domino type t satisfying the three conditions mentioned above.

Recurrent tile. As last task, we impose that the grid embedded into $\mathcal{G}$ has the distinguished domino type t^* occurring infinitely often in its first row. To do this, we first use two formulas that determine if a row or a column is the first one or not w.r.t. the orders $s_\alpha^{<}$ and $s_\beta^{<}$, respectively. Formally, we use $0_a(z) \triangleq \neg\langle\!\langle z' \rangle\!\rangle\, z' <_a z$, for $a \in \mathrm{Ag}$. One can prove that $\mathcal{G}, \chi, \varepsilon \models 0_\alpha(z)$ iff $\chi(z) \in F_0^a$.

Now, the infinite occurrence requirement on t^* can be expressed with the following sentence: $\varphi^{rec} \triangleq [\![x]\!][\![y]\!]\,(0_\beta(y) \wedge (0_\alpha(x) \vee (\alpha,x)(\beta,y)(\mathsf{X}t^*))) \to \langle\!\langle x' \rangle\!\rangle\, x <_\alpha x' \wedge (\alpha,x')(\beta,y)(\mathsf{X}t^*)$. Informally, φ^{rec} asserts that, when we are on the first row individuated by the variable y and at a column individuated by x such that it is the first column or the node of the "intersection" between x and y is labeled by t^*, we have that there exists a greater column individuated by x' such that its "intersection" with y is labeled by t^* as well.

Construction correctness. Now we have all the tools to formally prove the correctness of the undecidability reduction, by showing the equivalence between finding the solution of the recurrent tiling problem and the satisfiability of the sentence φ^{dom}. In particular, one can note that in the reduction we propose, only the $\mathrm{SL}_{\mathrm{CHP}}$ fragment of SL is involved. Thus, we prove that $\mathrm{SL}_{\mathrm{CHP}}$ under the concurrent semantics is highly undecidable, while it remains an open question whether this problem is undecidable or not in the turned-based framework.

▶ **Theorem 5.3.** *The satisfiability problem for* $\mathrm{SL}_{\mathrm{CHP}}$ *under the concurrent semantics is highly undecidable. In particular, it is* Σ^1_1-HARD.

References

1 R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-Time Temporal Logic. *JACM*, 49(5):672–713, 2002.

2 T. Brihaye, A.D.C. Lopes, F. Laroussinie, and N. Markey. ATL with Strategy Contexts and Bounded Memory. In *LFCS'09*, LNCS 5407, pages 92–106. Springer, 2009.

3 K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. In *CONCUR'07*, LNCS 4703, pages 59–73. Springer, 2007.

4 E.M. Clarke and E.A. Emerson. Design and Synthesis of Synchronization Skeletons Using Branching-Time Temporal Logic. In *LP'81*, LNCS 131, pages 52–71. Springer, 1981.

5 E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* MIT Press, 2002.

6 H.D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer, 1995.

7 E.A. Emerson and J.Y. Halpern. "Sometimes" and "Not Never" Revisited: On Branching Versus Linear Time. *JACM*, 33(1):151–178, 1986.

8 E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics, and Infinite Games: A Guide to Current Research.* LNCS 2500. Springer, 2002.

9 D. Harel. A Simple Highly Undecidable Domino Problem. In *LCC'84*, 1984.

10 W. Jamroga and W. van der Hoek. Agents that Know How to Play. *FI*, 63(2-3):185–219, 2004.

11 D. Kozen. Results on the Propositional mu-Calculus. *TCS*, 27:333–354, 1983.

12 O. Kupferman and M.Y. Vardi. Weak Alternating Automata and Tree Automata Emptiness. In *STOC'98*, pages 224–233, 1998.

13 O. Kupferman, M.Y. Vardi, and P. Wolper. An Automata Theoretic Approach to Branching-Time Model Checking. *JACM*, 47(2):312–360, 2000.

14 O. Kupferman, M.Y. Vardi, and P. Wolper. Module Checking. *IC*, 164(2):322–344, 2001.

15 D.E. Muller and P.E. Schupp. Simulating Alternating Tree Automata by Nondeterministic Automata: New Results and New Proofs of Theorems of Rabin, McNaughton and Safra. *TCS*, 141:69–107, 1995.

16 M. Pauly. A Modal Logic for Coalitional Power in Games. *JLC*, 12(1):149–166, 2002.

17 S. Pinchinat. A Generic Constructive Solution for Concurrent Games with Expressive Constraints on Strategies. In *ATVA'07*, LNCS 4762, pages 253–267. Springer, 2007.

18 A. Pnueli. The Temporal Logic of Programs. In *FOCS'77*, pages 46–57, 1977.

19 J.P. Queille and J. Sifakis. Specification and Verification of Concurrent Programs in Cesar. In *SP'81*, LNCS 137, pages 337–351. Springer, 1981.

20 S. Schewe. ATL* Satisfiability is 2ExpTime-Complete. In *ICALP'08*, LNCS 5126, pages 373–385. Springer, 2008.

21 M.Y. Vardi and P. Wolper. An Automata-Theoretic Approach to Automatic Program Verification. In *LICS'86*, pages 332–344. IEEE Computer Society, 1986.

22 H. Wang. Proving Theorems by Pattern Recognition II. *BSTJ*, 40:1–41, 1961.